Package: AGHmatrix (via r-universe)

October 31, 2026

Type Package

Title Relationship Matrices for Diploid and Autopolyploid Species

Version 3.0.3

Description Fast computation of A (pedigree), G (genomic-base), and H (A corrected by G) relationship matrices for diploid and autopolyploid species. Several methods are implemented considering additive and non-additive models.

Maintainer Rodrigo Amadeu <rramadeu@gmail.com>

Depends R (>= 3.6.0)

Imports Rcpp (>= 1.1.0), matrixStats (>= 1.5.0)

Suggests knitr, MASS, rmarkdown, testthat, zoo

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Encoding UTF-8

License GPL-3

URL https://github.com/rramadeu/AGHmatrix

RoxygenNote 7.3.2

NeedsCompilation yes

Author Rodrigo Amadeu [aut, cre], Luis Ferrao [aut, ctb], Thiago Oliveira [aut, ctb], Catherine Cellon [ctb], Leticia Lara [ctb], Marcio Resende [ctb], Ivone Oliveira [ctb], Patricio Munoz [ctb], Augusto Garcia [ctb]

Repository https://cran.r-universe.dev

Date/Publication 2026-10-01 19:37:48 UTC

RemoteUrl https://github.com/cran/AGHmatrix

RemoteRef HEAD

RemoteSha 4c38bae213e439948d012b7fba725dbee0111565

Contents

Amatrix 2
AmatrixPolyCross 4
asciitonumber 6
datatreat 6
expandAmatrix 7
filterpedigree 8
formatmatrix 9
Gmatrix 10
H_martini_blocks 13
Hmatrix 13
Mcheck 15
missingdata 17
munoz_var_mean_by_Aclass 17
ped.mrode 18
ped.sol 19
snp.pine 19
snp.sol 20

Index **21**

`Amatrix` *Construction of Relationship Matrix A*

Description

Constructs an additive relationship matrix (A) from pedigree data in a three-column format. The calculation depends on the specified ploidy level (must be an even number). If `ploidy = 4`, the matrix can incorporate a user-defined proportion (`w`) of parental gametes that are IBD (Identical by Descent) due to double reduction.

When `ploidy = 2` and `dominance = FALSE`, the diploid additive numerator relationship matrix is built using Henderson's (1976) method. If `dominance = TRUE`, a diploid dominance numerator relationship matrix is computed following Cockerham (1954). The recursive approach is detailed in Mrode (2005).

For `ploidy > 2` and `dominance = FALSE`, an autopolyploid additive relationship matrix is computed based on Kerr et al. (2012). If `slater = TRUE` and `ploidy = 4`, Slater's (2013) method is used instead.

Usage

```
Amatrix(
  data = NULL,
  ploidy = 2,
  w = 0,
  verify = TRUE,
  dominance = FALSE,
```

```
  slater = FALSE,
  ASV = FALSE,
  ...
)
```

Arguments

`data`	Pedigree data frame in a 3-column format. Unknown values should be set to 0.
`ploidy`	An even number representing the ploidy level (default is 2).
`w`	Proportion of parental gametes that are IBD due to double reduction (default is 0). Applicable only when `ploidy = 4`.
`verify`	Logical; if TRUE (default), the pedigree file is checked for conflicting entries.
`dominance`	Logical; if TRUE, computes the dominance relationship matrix (valid only for `ploidy = 2`).
`slater`	Logical; if TRUE and `ploidy = 4`, computes the additive autotetraploid relationship matrix using Slater (2013).
`ASV`	Logical; if TRUE, transforms the matrix into the average semivariance (ASV) defined as $K = K/(\text{trace}(K)/(\text{nrow}(K) - 1))$. See Feldmann et al. (2022), Formula 2.
`...`	Additional arguments passed to `datatreat()`.

Value

A square matrix representing the relationship between individuals in the pedigree.

Note

This function uses compiled C++ code via Rcpp for improved performance.

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Thiago de Paula Oliveira, <toliveira@abacusbio.com>

References

Cockerham, C.C. (1954). An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. Genetics, 39, 859–882.

Feldmann, M.J., et al. (2022). Average semivariance directly yields accurate estimates of the genomic variance in complex trait analyses. G3 (Bethesda), 12(6).

Henderson, C.R. (1976). A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics, 32, 69–83.

Kerr, R.J., et al. (2012). Use of the numerator relationship matrix in genetic analysis of autopolyploid species. Theoretical and Applied Genetics, 124, 1271–1282.

Mrode, R.A. (2014). Linear Models for the Prediction of Animal Breeding Values (3rd ed.). CABI.

Slater, A.T., et al. (2013). Improving the analysis of low heritability complex traits for enhanced genetic gain in potato. Theoretical and Applied Genetics, 127, 809–820.

Examples

```
data(ped.mrode)
# Additive relationship matrix for diploids (Henderson 1976):
Amatrix(ped.mrode, ploidy = 2)

# Dominance relationship matrix for diploids (Cockerham 1954):
Amatrix(ped.mrode, ploidy = 2, dominance = TRUE)

# Additive relationship matrix for autotetraploids (Kerr 2012):
Amatrix(ped.mrode, ploidy = 4)

# Additive relationship matrix for autooctaploids (Kerr 2012):
Amatrix(ped.mrode, ploidy = 8)

# Additive relationship matrix for autotetraploids with double reduction
# (Kerr 2012):
Amatrix(ped.mrode, ploidy = 4, w = 0.1)

# Additive relationship matrix for autotetraploids with double reduction
# using Slater (2013):
Amatrix(ped.mrode, ploidy = 4, w = 0.1, slater = TRUE)

# Additive relationship matrix for autohexaploids with double reduction
# (Kerr 2012):
Amatrix(ped.mrode, ploidy = 6, w = 0.1)
```

AmatrixPolyCross *Construction of pedigree-based relationship matrix with parental guessing possibility*

Description

Creates an additive relationship matrix A based on a non-deterministic pedigree with 4+ columns where each column represents a possible parent. This function was built with the following designs in mind. 1) A mating design where you have equally possible parents. For example, a generation of insects derived from the mating of three insects in a cage. All the insects in this generation will have the same expected relatedness with all the possible parents (1/3). If there are only two parents in the cage, the function assumes no-inbreeding and the pedigree is deterministic (the individual is offspring of the cross between the two parents). Another example, a population of 10 open-pollinated plants where you harvest the seeds without tracking the mother. 2) When fixedParent is TRUE: a mating design where you know one parent and might know the other possible parents. For example, a polycross design where you have seeds harvested from a mother plant and possible polen donors.

Usage

```
AmatrixPolyCross(data = NULL, fixedParent = FALSE)
```

Arguments

data	pedigree data name. Unknown value should be equal 0. See example for construction.
fixedParent	if false, assumes that all the parents are equally possible parents. If true, assumes that the first parental is known and the others are equally possible parents. Default = FALSE.

Value

Matrix with the relationship between the individuals.

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Examples

```
#the following pedigree has the id of the individual followed by possible parents
#if 0 is unknown
#the possible parents are filled from left to right
#in the pedigree data frame examples:
#id 1,2,3,4 have unknown parents and are assumed unrelated
#id 5 has three possible parents (1,2,3)
#id 6 has three possible parents (2,3,4)
#id 7 has two parents (deterministic case here, the parents are 3 and 4)
#id 8 has four possible parents (5,6,7,1)

pedigree = data.frame(id=1:8,
                      parent1 = c(0,0,0,0,1,2,3,5),
                      parent2 = c(0,0,0,0,2,3,4,6),
                      parent3 = c(0,0,0,0,3,4,0,7),
                      parent4 = c(0,0,0,0,0,0,0,1),
                      parent5 = 0)

print(pedigree)

AmatrixPolyCross(pedigree)

#when polyCross is set to be true:
#id 5 is offspring of parent 1 in a deterministic way and two other possible parents (2,3)
#id 6 is offspring of parent 2 in a deterministic way and two other possible parents (3,4)
#id 7 has two parents (deterministic case here, the parents are 3 and 4); as before
#id 8 is offspring of parent 5 in a deterministic way and has three other possible parents (6,7,1)

AmatrixPolyCross(pedigree,fixedParent=TRUE)
```

asciitonumber *Convert pedigree names to numeric indices using C++*

Description

Convert pedigree names to numeric indices using C++

Usage

```
asciitonumber(pedigree.data, unk = 0)
```

Arguments

pedigree.data	A 3-column matrix or data frame with individual, sire, and dam
unk	Code for unknown parent (default = "0")

Value

A list with sire, dam, and individual names

Author(s)

Rodrigo R. Amadeu <rramadeu@gmail.com>

Thiago de Paula Oliveira

datatreat *Organizes pedigree data in a chronological way using C++*

Description

This function organizes pedigree data in a chronological way and returns a list: i) parental 1 values (numeric); ii) parental 2 values (numeric); iii) individual names. Also saves a '.txt' file if 'save = TRUE'.

Usage

```
datatreat(data = NULL, n.max = 50, unk = 0, save = FALSE)
```

Arguments

data	A 3-column data frame or matrix with individual, sire, and dam
n.max	Max number of iterations (default = 50)
unk	Code for unknown parent (default = 0)
save	If TRUE, saves the reordered pedigree to a text file

Value

A list with elements: sire (numeric), dire (numeric), ind_data (names)

Author(s)

Rodrigo R. Amadeu <rramadeu@gmail.com>

Thiago de Paula Oliveira

Examples

```
# data(ped.mrode)
# datatreat(ped.mrode)
```

`expandAmatrix` *Add new crosses to a current A matrix*

Description

Expand a current A matrix with a new pedigree. The parents in the new pedigree should also be in the A matrix.

Usage

```
expandAmatrix(newPedigree = NULL, A = NULL, returnAll = TRUE)
```

Arguments

`newPedigree`	pedigree data name (3-column way format). Unknown value should be equal 0.
`A`	numerator relationship matrix output from Amatrix function.
`returnAll`	if TRUE returns old A with new A, if FALSE returns only new A

Value

Matrix with the Relationship between the individuals.

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Examples

```
data(ped.sol)
ped.initial = ped.sol[1:1120,]
ped.new = ped.sol[-c(1:1120),]
#Computing additive relationship matrix:
A = Amatrix(ped.initial, ploidy=2)
Anew = expandAmatrix(ped.new, A)

#Comparing with one-step building..
Afull = Amatrix(ped.sol, ploidy=2)
test = Anew-Afull
which(test!=0)
```

`filterpedigree` *Filter the pedigree to keep only the genealogy of a subset of individuals*

Description

Filter the pedigree to keep only the genealogy of a subset of individuals

Usage

```
filterpedigree(inds, data)
```

Arguments

`inds`	vector with strings of individuals to keep their genealogy in the matrix
`data`	name of the pedigree data frame. Default=NULL.

Value

a data frame with pedigree containing the genealogy of the selected individuals

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Examples

```
data(ped.sol)
new.ped.sol = filterpedigree(inds = c("MSW168-2","W14090-3","W14090-4"),data=ped.sol)
```

formatmatrix *Transform a matrix in 3 columns*

Description

Given any square matrix transform it in a 3 columns way (row, column, value) mainly to be used in outsourcing data processing (as ASREML-standalone)

Usage

```
formatmatrix(
  data = NULL,
  save = TRUE,
  return = FALSE,
  name = deparse(substitute(data)),
  round.by = 12,
  exclude.0 = TRUE
)
```

Arguments

data	matrix (nxn).
save	if TRUE save the output in a file. Default=TRUE.
return	if TRUE return the output in a object. Default=FALSE.
name	name of the csv file to be saved. Default=data name.
round.by	select the number of digits after 0 you want in your data. Default = 12
exclude.0	if TRUE, remove all lines equal to zero (ASREML option). Default = TRUE

Value

a object or a csv file with a table with 3 columns representing the matrix.

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Examples

```
#Example with random matrix
data<-matrix(c(1,0.1,0,0.1,1,0,0,0,1.1),3)
formatmatrix(data=data,save=FALSE,return=TRUE,exclude.0=TRUE)

#Example with pedigree matrix
#Reading the example data
data(ped.mrode)
#Making Relationship Matrix
Amrode<-Amatrix(ped.mrode)
```

```
#Inverting the Matrix
Amrode.inv<-solve(Amrode)
#Making the 3 columns format
Amrode.inv.ASREML<-formatmatrix(Amrode,save=FALSE,return=TRUE,exclude.0=TRUE)
#Printing it
Amrode.inv.ASREML
```

`Gmatrix` *Construction of Relationship Matrix G*

Description

Given a matrix (individual x markers), a method, a missing value, and a maf threshold, return a additive or non-additive relationship matrix. For diploids, the methods "Yang" and "VanRaden" for additive relationship matrices, and "Su" and "Vitezica" for non-additive relationship matrices are implemented. For autopolyploids, the method "VanRaden" for additive relationship, method "Slater" for full-autopolyploid model including non-additive effects, and pseudo-diploid parametrization are implemented. Weights are implemented for "VanRaden" method as described in Liu (2020).

Usage

```
Gmatrix(
  SNPmatrix = NULL,
  method = "VanRaden",
  missingValue = -9,
  maf = 0,
  thresh.missing = 0.5,
  verify.posdef = FALSE,
  ploidy = 2,
  pseudo.diploid = FALSE,
  integer = TRUE,
  ratio = FALSE,
  impute.method = "mean",
  rmv.mono = FALSE,
  thresh.htzy = 0,
  ratio.check = TRUE,
  weights = NULL,
  ploidy.correction = FALSE,
  ASV = FALSE
)
```

Arguments

`SNPmatrix`	matrix (n x m), where n is is individual names and m is marker names (coded inside the matrix as 0, 1, 2, ..., ploidy, and, missingValue).

`method` "Yang" or "VanRaden" for marker-based additive relationship matrix. "Su" or "Vitezica" for marker-based dominance relationship matrix. "Slater" for full-autopolyploid model including non-additive effects. "Endelman" for autotetraploid dominant (digentic) relationship matrix. "MarkersMatrix" for a matrix with the amount of shared markers between individuals (3). Default is "VanRaden", for autopolyploids will be computed a scaled product (similar to Covarrubias-Pazaran, 2006).

`missingValue` missing value in data. Default=-9.

`maf` minimum allele frequency accepted to each marker. Default=0.

`thresh.missing` threshold on missing data, SNPs below of this frequency value will be maintained, if equal to 1, no threshold and imputation is considered. Default = 1.

`verify.posdef` verify if the resulting matrix is positive-definite. Default=FALSE.

`ploidy` data ploidy (an even number between 2 and 20). Default=2.

`pseudo.diploid` if TRUE, uses pseudodiploid parametrization of Slater (2016).

`integer` if FALSE, not check for integer numbers. Default=TRUE.

`ratio` if TRUE, molecular data are considered ratios and its computed the scaled product of the matrix (as in "VanRaden" method).

`impute.method` "mean" to impute the missing data by the mean per marker, "mode" to impute the missing data by the mode per marker, "global.mean" to impute the missing data by the mean across all markers, "global.mode" to impute the missing data my the mode across all marker. Default = "none".

`rmv.mono` if monomorphic markers should be removed. Default=FALSE.

`thresh.htzy` threshold heterozigosity, remove SNPs below this threshold. Default=0.

`ratio.check` if TRUE, run Mcheck with ratio data.

`weights` vector with weights for each marker. Only works if method="VanRaden". Default is a vector of 1's (equal weight).

`ploidy.correction`
It sets the denominator (correction) of the crossprod. Used only when ploidy > 2 for "VanRaden" and ratio models. If TRUE, it uses the sum of "Ploidy" times "Frequency" times "(1-Frequency)" of each marker as method 1 in VanRaden 2008 and Endelman (2018). When ratio=TRUE, it uses "1/Ploidy" times "Frequency" times "(1-Frequency)". If FALSE, it uses the sum of the sampling variance of each marker. Default = FALSE.

`ASV` if TRUE, transform matrix into average semivariance (ASV) equivalent (K = K / (trace(K) / (nrow(K)-1))). Details formula 2 of Fieldmann et al. (2022). Default = FALSE.

Value

Matrix with the marker-bases relationships between the individuals

Author(s)

Rodrigo R. Amadeu <rramadeu@gmail.com>

Thiago de Paula Oliveira <toliveira@abacusbio.com>

Marcio Resende Jr.

Letícia A. C. Lara

Ivone Oliveira

Felipe V. Ferrao

References

Covarrubias-Pazaran, G. 2016. Genome assisted prediction of quantitative traits using the R package sommer. PLoS ONE 11(6):1-15.

Endelman, JB, et al., 2018. Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. Genetics, 209(1) pp. 77-87.

Feldmann MJ, et al. 2022. Average semivariance directly yields accurate estimates of the genomic variance in complex trait analyses. G3 (Bethesda), 12(6).

Liu, A, et al. 2020. Weighted single-step genomic best linear unbiased prediction integrating variants selected from sequencing data by association and bioinformatics analyses. Genet Sel Evol 52, 48.

Slater, AT, et al. 2016. Improving genetic gain with genomic selection in autotetraploid potato. The Plant Genome 9(3), pp.1-15.

Su, G, et al. 2012. Estimating additive and non-additive genetic variances and predicting genetic merits using genome-wide dense single nucleotide polymorphism markers. PloS one, 7(9), p.e45293.

VanRaden, PM, 2008. Efficient methods to compute genomic predictions. Journal of dairy science, 91(11), pp.4414-4423.

Vitezica, ZG, et al. 2013. On the additive and dominant variance and covariance of individuals within the genomic selection scope. Genetics, 195(4), pp.1223-1230.

Yang, J, et al. 2010. Common SNPs explain a large proportion of the heritability for human height. Nature genetics, 42(7), pp.565-569.

Examples

```
## Not run:
## Diploid Example
data(snp.pine)
str(snp.pine)
Gmatrix.Yang <- Gmatrix(snp.pine, method="Yang", missingValue=-9, maf=0.05)
Gmatrix.VanRaden <- Gmatrix(snp.pine, method="VanRaden", missingValue=-9, maf=0.05)
Gmatrix.Su <- Gmatrix(snp.pine, method="Su", missingValue=-9, maf=0.05)
Gmatrix.Vitezica <- Gmatrix(snp.pine, method="Vitezica", missingValue=-9, maf=0.05)

## Autetraploid example
data(snp.sol)
Gmatrix.VanRaden <- Gmatrix(snp.sol, method="VanRaden", ploidy=4)
```

```
Gmatrix.Endelman <- Gmatrix(snp.sol, method="Endelman", ploidy=4)
Gmatrix.Slater <- Gmatrix(snp.sol, method="Slater", ploidy=4)
Gmatrix.Pseudodiploid <- Gmatrix(snp.sol, method="VanRaden", ploidy=4, pseudo.diploid=TRUE)
Gmatrix.weighted <- Gmatrix(snp.sol, method="VanRaden", weights = runif(3895,0.001,0.1), ploidy=4)

## End(Not run)
```

H_martini_blocks *Compute Martini block corrections (H11, H12, H21, H22 - A22)*

Description

Compute Martini block corrections (H11, H12, H21, H22 - A22)

Usage

```
H_martini_blocks(A12, A22, G22, tau, omega)
```

Arguments

A12	n1 x n2 block from A
A22	n2 x n2 block from A (genotyped)
G22	n2 x n2 block from G (genotyped)
tau	scalar
omega	scalar

Value

list(H11, H12, H21, H22corr) where H22corr = H22 - A22

Hmatrix *Construction of Combined Relationship Matrix H*

Description

Given a matrix A and a matrix G returns a H matrix. H matrix is the relationship matrix using combined information from the pedigree and genomic relationship matrices. First, you need to compute the matrices separated and then use them as input to build the combined H matrix. Two methods are implemented: 'Munoz' shrinks the G matrix towards the A matrix scaling the molecular relatadness by each relationship classes; 'Martini' is a modified version from Legarra et al. (2009) where combines A and G matrix using scaling factors. When method is equal 'Martini' and 'tau=1' and 'omega=1' you have the same H matrix as in Legarra et al. (2009).

Usage

```
Hmatrix(
  A = NULL,
  G = NULL,
  markers = NULL,
  c = 0,
  method = "Martini",
  tau = 1,
  omega = 1,
  missingValue = -9,
  maf = 0,
  ploidy = 2,
  roundVar = 3,
  ASV = FALSE
)
```

Arguments

A	A matrix from function Amatrix
G	G matrix from function Gmatrix
markers	matrix marker which generated the Gmatrix
c	constant value of H computation, default: c=0
method	"Martini" or "Munoz", default="Martini"
tau	to be used for Martini's method, default=1.
omega	to be used of Martini's method, default=1.
missingValue	missing value in data, default=-9.
maf	max of missing data accepted to each markerm default=0.05.
ploidy	data ploidy (an even number between 2 and 20), default=2.
roundVar	only used for Munoz's method, how many digits to consider the relationship be of same class, default=2.
ASV	if TRUE, transform matrix into average semivariance (ASV) equivalent (K = K / (trace(K) / (nrow(K)-1))). Details formula 2 of Fieldmann et al. (2022). Default = FALSE.

Value

H Matrix with the relationship between the individuals based on pedigree and corrected by molecular information

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Thiago de Paula Oliveira <toliveira@abacusbio.com>

References

Feldmann MJ, et al. 2022. Average semivariance directly yields accurate estimates of the genomic variance in complex trait analyses. G3 (Bethesda), 12(6).

Munoz, PR. 2014 Unraveling additive from nonadditive effects using genomic relationship matrices. Genetics 198, 1759-1768

Martini, JW, et al. 2018 The effect of the H-1 scaling factors tau and omega on the structure of H in the single-step procedure. Genetics Selection Evolution 50(1), 16

Legarra, A, et al. 2009 A relationship matrix including full pedigree and genomic information. Journal of Dairy Science 92, 4656-4663

Examples

```
## Not run:
data(ped.sol)
data(snp.sol)
#Computing the numerator relationship matrix 10% of double-reduction
Amat <- Amatrix(ped.sol, ploidy=4, w = 0.1)
#Computing the additive relationship matrix based on VanRaden (modified)
Gmat <- Gmatrix(snp.sol, ploidy=4,
                maf=0.05, method="VanRaden")
Gmat <- round(Gmat,3) #to be easy to invert

#Computing H matrix (Martini)
Hmat_Martini <- Hmatrix(A=Amat, G=Gmat, method="Martini",
                    ploidy=4,
                    maf=0.05)

#Computing H matrix (Munoz)
Hmat_Munoz <- Hmatrix(A=Amat, G=Gmat, markers = snp.sol,
                     ploidy=4, method="Munoz",
                     roundVar=2,
                     maf=0.05)

## End(Not run)
```

Mcheck *Check and filter markers*

Description

This function does different filtering on the marker matrix

Usage

```
Mcheck(
  SNPmatrix = NULL,
```

```
  ploidy = 2,
  missingValue = -9,
  thresh.maf = 0.05,
  thresh.missing = 0.9,
  thresh.htzy = 0,
  impute.method = "mean",
  rmv.mono = TRUE
)
```

Arguments

SNPmatrix	matrix (n x m), where n is is individual names and m is marker names (coded inside the matrix as 0, 1, 2, ..., ploidy, and, missingValue).
ploidy	data ploidy (an even number between 2 and 20). Default=2.
missingValue	missing value in data. Default=-9.
thresh.maf	minimum allele frequency accepted to each marker. Default=0.05.
thresh.missing	threshold on missing data, SNPs below of this frequency value will be maintained, if equal to 1, no threshold and imputation is considered. Default = 0.50.
thresh.htzy	threshold heterozigosity, remove SNPs below this threshold. Default=0.
impute.method	"mean" to impute the missing data by the mean per marker, "mode" to impute the missing data by the mode per marker, "global.mean" to impute the missing data by the mean across all markers, "global.mode" to impute the missing data my the mode across all marker. Default = "mean".
rmv.mono	if monomorphic markers should be removed. Default=TRUE.

Value

SNPmatrix after filtering steps.

Author(s)

Luis F V Ferrao

Rodrigo Amadeu, <rramadeu@gmail.com>

Thiago de Paula Oliveira <toliveira@abacusbio.com>

Examples

```
data(snp.pine)
M = Mcheck(snp.pine)
```

missingdata *Survying on missing data*

Description

This function verify which rows in a pedigree data has missing parental or conflictuos data

Usage

```
missingdata(data, unk = 0)
```

Arguments

data	data name from a pedigree list
unk	unknown value of your data

Value

list with $conflict: rows of the data which are at least one parental name equal to the individual. $missing.sire: rows of the data which arie missing data sire (Parental 1) information. $missing.dire: same as above for dire (Parental 2). $summary.missing: summary of the missing data. 2 columns, 1st for the name of the parental listed, 2nd for the how many times appeared in the data.

Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

Examples

```
data(ped.mrode)
missingdata(ped.mrode)
```

munoz_var_mean_by_Aclass *Compute var/mean of G by classes of rounded A, and fill varA/meanG*

Description

Compute var/mean of G by classes of rounded A, and fill varA/meanG

Usage

```
munoz_var_mean_by_Aclass(G, A, round_digits)
```

Arguments

G	square matrix
A	square matrix (same dim as G)
round_digits	integer number of decimal places to round A

Value

list(varA, meanG, classes, var_by_class, mean_by_class)

---

ped.mrode *Pedigree Data*

Description

Data from pedigree example proposed by Mrode 2005

Usage

```
data(ped.mrode)
```

Format

table

References

R. A. Mrode, R. Thompson. Linear Models for the Prediction of Animal Breeding Values. CABI, 2005.

Examples

```
data(ped.mrode)
```

ped.sol *Pedigree data for autopolyploid examples*

Description

Dataset extract from supplementary material from Endelman et al. (2018). Pedigree data frame of Potato population, missing data as 0.

Usage

```
data(ped.sol)
```

Format

data.frame

References

Endelman, JB, et al., 2018 Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. Genetics, 209(1) pp. 77-87.

Examples

```
data(ped.sol)
```

snp.pine *Molecular data for diploid examples*

Description

Dataset extract from supplementary material from Resende et al. (2012). SNP marker matrix from Pine tree coded as 0,1, and 2, and missing value as -9.

Usage

```
data(snp.pine)
```

Format

matrix

References

Resende, MF, et al., 2012 Accuracy of genomic selection methods in a standard data set of loblolly pine (Pinus taeda l.). Genetics 190: 1503–1510.

Examples

```
data(snp.pine)
```

`snp.sol` *Molecular data for autopolyploid examples*

Description

Dataset extract from supplementary material from Endelman et al. (2018). SNP marker matrix from Pine tree coded as 0,1,2,3,4 and missing value as -9.

Usage

```
data(snp.sol)
```

Format

data.frame

References

Endelman, JB, et al., 2018 Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. Genetics, 209(1) pp. 77-87.

Examples

```
data(snp.sol)
```

Index

∗ **datasets**
ped.mrode, 18
ped.sol, 19
snp.pine, 19
snp.sol, 20

Amatrix, 2
AmatrixPolyCross, 4
asciitonumber, 6

datatreat, 6

expandAmatrix, 7

filterpedigree, 8
formatmatrix, 9

Gmatrix, 10

H_martini_blocks, 13
Hmatrix, 13

Mcheck, 15
missingdata, 17
munoz_var_mean_by_Aclass, 17

ped.mrode, 18
ped.sol, 19

snp.pine, 19
snp.sol, 20